#6

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Mark Phillips, Jonathan Cook, Pericles Haleftiras, Lizanne Kaiser, Jon Thomas Layton		
Assignee:	General Magic Inc.		
Title:	VOICE APPLICATION DEVELOPMENT METHODOLOGY		
Serial No.:	Unassigned	Filing Date:	December 14, 2001
Examiner:	Unassigned	Group Art Unit:	Unassigned
Docket No.:	M-7199-5P US		

San Jose, California
December 28, 2001

COMMISSIONER FOR PATENTS
Washington, D. C. 20231

PRELIMINARY AMENDMENT

Dear Sir:

The following Amendments and Remarks are submitted for entry into the continuation-in-part Application, which was filed from application serial number 09/855,004 filed May 14, 2001.

AMENDMENTS

Please amend the above-referenced application as follows:

01/18/2002 MBERHE 00000033 192386 10017435
01 FC:103 138.00 CH

In the Specification

Please replace the first paragraph on page 1 with the following rewritten paragraph:

The present invention relates generally to voice applications and, more particularly, to a methodology of using a design methodology platform, having generic software components, and a deployment environment to develop and deploy a specific voice application.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the fourth paragraph on page 2 with the following rewritten paragraph:

After the script is developed, a grammar is developed during a grammar development phase. In voice applications that do not use Dual Tone Multi-Frequency (DTMF) as the primary caller communication mechanism, Automatic Speech Recognition (ASR) is used to recognize what a caller has spoken and to translate that into the action that the caller wishes to take. The recognition rules for performing these functions are captured within an ASR grammar, which maps a set of allowed utterances to a set of appropriate actions or interpretations. Grammars are specific, i.e., different ASR engines may process rules in different ways.

Please replace the last paragraph on page 2 with the following rewritten paragraph:

After a grammar is developed then prompts are developed during a prompt development phase. When interacting with a caller, a voice application should speak to the caller. It is known to speak to a caller using Text-to-Speech (TTS) technology or using pre-recorded speech. In the latter case, a speech talent records phrases that are required as defined within a dialog design specification. It is usual for these prompts to be recorded in an audio format that is not optimized for the telephony environment. Often processing is used to convert the audio format. This conversion is generally performed by a sound processing engine. Also, audio files are often concatenated to build full phrases used by an application as independently recording every utterance would be inefficient.

Please replace the third paragraph on page 3 with the following rewritten paragraph:

Once tested then the voice application is ready to be deployed. Applications must often be deployed on multiple platforms. For example, parts of an application may be

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

deployed to a script server, a voice gateway, an ASR server, etc. Often when deployed, the application environment is replicated for redundancy or scalability.

Please replace the third paragraph on page 5 with the following rewritten paragraph:

FIGURE 21 is a flow diagram of an exemplary method 200 of operation for **a** remote system.

Please replace the last paragraph on page 6 with the following rewritten paragraph:

Referring to FIGURE 1, a voice application is developed in accordance with a design methodology and, in the preferred embodiment, is developed using a design methodology platform 2 (FIGURE 4). The design methodology includes a design phase 1102 and a deployment phase 1104. That is, once a specific voice application is developed and tested in the design phase 1102, the voice application is deployed at deployment phase 1104. Once the specific voice application is deployed at deployment phase 1104, then the specific voice application is maintained in an iterative fashion as illustrated in FIGURE 1. The regular ongoing maintenance improves voice recognition accuracy and usability. The maintenance includes analyzing data from the specific voice application and making adjustments to the dialog flow and grammars to improve the user's overall user experience with the specific voice application.

Please replace the first paragraph on page 7 with the following rewritten paragraph:

Referring to FIGURE 2, the design phase 1102 of the methodology includes the utilization of a plurality of generic software components in order to develop a specific voice application. Such generic software components comprise a design methodology platform 2 (FIGURE 4). More specifically, the design phase 1102 of the voice application design methodology includes utilization of various components of the platform 2. The design phase

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

1102 of the methodology includes a dialog design phase 1202, a voice coding phase 1204, a personalization phase 1206, a custom prompt development phase 1208, a custom grammar development phase 1210, a standard prompt phase 1212, a standard grammar phase 1214, and a system test phase 1216.

Please replace the second paragraph on page 7 with the following rewritten paragraph:

The dialog design phase 1202 involves designing how the user is expected to interact with the voice application that will be deployed. For example, the dialog design phase 1202 may include the definition of prompts that are designed to elicit a predictable response from the user. Dialog flows may be generated, the dialog flows capturing and defining the anticipated scenarios of user interaction with the voice application that is to be deployed. These dialog flows are developed into scripts. In this manner, during the dialog design phase 1202, a voice application developer defines the dialog for the specific voice application under development.

Please replace the second paragraph on page 8 with the following rewritten paragraph:

In the prompt development phases 1208, 1212, the developer of the voice application generates utterances, or prompts, that will be spoken by the voice application to the user. These prompts can be generated using Text-to-speech (TTS) technology and can also be generated using pre-recorded speech. For a voice application that is generated using predesigned generic prompts, phase 1212 is executed. In contrast, custom prompts can be generated in the speech prompts recording phase 1208. In phase 1212, generic pre-generated prompts are used. In phase 1208, a speech talent records prompts as defined by the scripts generated in the dialog design phase 1202.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the third paragraph on page 8 with the following rewritten paragraph:

The design methodology also includes grammar development phases 1210, 1214, wherein the developer of the voice application generates grammars. Grammars define which user utterances the voice application recognizes, and provides that the user utterances be translated into a determination of what action the user would like to invoke in the voice application. A grammar is a set of recognition rules. A grammar maps a set of allowed utterances to a set of appropriate actions (or interpretations). One or more grammars for a voice application may be generated using the standard grammar components in phase 1214. Similarly, one or more grammars for a voice application may be generated using custom grammar development in phase 1210.

Please replace the third paragraph on page 9 with the following rewritten paragraph:

FIGURES 2 and 3 illustrate that generic components 310 are available for the voice application developer's use during the standard prompts 1212 and standard grammar 1214 phases of the design phase 1102. In contrast, custom prompts and grammars are created in phases 1208 and 1210 following the method illustrated in operations 312 through 316 in FIGURE 3. The custom components are coded by the voice application designer in operation 312. The custom-coded components are then tested in a standalone environment in operation 314. If tested successfully, the custom-coded components are then integrated with the voice application. In operation 318, the completed application is tested, regardless of whether the application used standard components or custom-coded components.

Please replace the sixth paragraph on page 14 with the following rewritten paragraph:

The component data access layer 810 is an optional part of the component 540, dependent on the functionality required, and is the interface to the back-end resources that are

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

required by the component 540. The component access data layer 810 utilizes the voice application services layer 3 to retrieve information from the various back-end data sources and services.

Please replace the last paragraph on page 14 with the following rewritten paragraph:

FIGURE 9 illustrates a component load sequence. FIGURE 9 illustrates that a component is invoked in sequence 902 when a request is issued by the voice gateway 4. The request is passed to the dialog control 525 with the URI of the component 540 to be loaded. The dialog control 525 performs a look-up in the dialog control configuration file 530. The dialog control 525 then fetches the appropriate component 540 from the component definition file 802 in sequence 906 and loads the appropriate component in sequence 908. FIGURE 9 illustrates that, in the remaining sequences 909-942, the dialog control 525 forwards the HTTP request to the component 540, passing the parameters from the request. The component script 804 then performs the function that is required by the application, including accessing any back-end services that are required.

Please replace the first paragraph on page 15 with the following rewritten paragraph:

FIGURE 10 illustrates the architecture of the prompt engine 1000 and associated prompt component. In addition to dialog components 540 the deployment environment 500 supports the invocation of prompt components 1108 by the voice application designer during phase 1212. These are reusable snippets of dynamic code that produce the prompting that is used to communicate with the caller. Each prompt component 1108 is defined and stored within the voice application repository 600 and can be executed by passing a set of input parameters. Prompts are defined within a file, such as an XML file. FIGURE 10 illustrates

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

that, in at least one embodiment, a runtime prompt engine 1000 loads the prompt file from the repository 600 and executes the prompt.

Please replace the first paragraph on page 16 with the following rewritten paragraph:

FIGURE 5 illustrates that the messaging services layer 539 allows voice applications 27 to send message requests to external systems 542a, 542b, 542n. In at least one embodiment, the messaging services layer 539 is a set of classes that provide the capability to create and bind data messages and send them to specific destinations through utilization of a number of various protocols. The messaging services provided by the messaging services layer 539 are accessible via the voice application service layer 3. The voice application service layer 3 provides a set of custom tag libraries that provide for interaction with the messaging services from within the voice application 27 (FIGURE 4). In at least one embodiment, the voice application service layer 3 provides a tag library to support the following message protocols: Java™ messaging service ("JMS") and Java™ XML messaging service ("JAXM").

Please replace the last paragraph on page 17 with the following rewritten paragraph:

In such cases (i.e., when a message is expired) , the messaging controller 1220 performs data-expiration monitoring and cleanup. If the application 27 requests to receive a message and there is no message available, then the messaging controller 1220 notifies the application 27 to wait for a specified timeout period. If the requested message does not become available during the timeout period, then the request becomes "timed-out." In such case, the messaging controller 1220 provides a negative acknowledgement to the message source 542 to indicate the requested message has not been delivered.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the second paragraph on page 18 with the following rewritten paragraph:

FIGURE 5 illustrates that another set of custom tags is provided to provide access to a rules integration layer 537. This set of tags, referred to as a personalization tag library, allows rules to be set up in a rules engine associated with the rules integration layer 537. This allows a rules set to be invoked and also allows for actions to be taken in the dialog based on the result of invocation of the rules set. The tags in the personalization tag library also provide access to information being tracked by the application. The custom tags allow the application developer to execute a rule set on a number of parameters to decide what actions should be performed by the application. Some appropriate uses for this interface are:

Please replace the first paragraph on page 19 with the following rewritten paragraph:

FIGURE 15 illustrates the architecture for the rules integration layer 537. The rules integration layer 537 includes a remote interface 1502 that facilitates communication with the voice application services layer 3. The rules integration layer 537 includes a set of custom tags that perform specific rule-type functions and includes a rules engine 1504. A rule engine pool manager 1506 performs initialization functions by polling and sharing available rule engine resources. Remote users assert objects into the context of the rule engine 1504 and then execute one or more rule files 1508. The rules service remote interface 1502 maintains state across method calls, allowing multiple objects to be asserted before rules are invoked. Once the rules have been executed, the rules integration layer 537 returns all objects that were affected by the execution of the rules. In at least one embodiment, the affected objects are returned as an object array.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the third paragraph on page 19 with the following rewritten paragraph:

FIGURE 5 illustrates that another set of custom tags is provided to provide access to a call detail tracking layer 541. The custom tags, referred to as a tracking tag library, provide for tracking of information about a caller session while the application is executing. To do this, the call detail tracking layer 541 provides for tracking information falling into a number of distinct categories, including call-based information, caller-based information, and event-based information. Call-based tracking involves tracking information about a particular call, including when the call started, dialed number, etc. Caller-based tracking involves tracking information about caller. This tracking feature is used in conjunction with a user profile in order to provide personalized content to the caller. Event-based tracking involves tracking the events that occur within the call flow. For example, event-based tracking will record that a caller has been transferred out of the specific voice application. As another example, event-based tracking will record that a caller has provided an invalid credential.

Please replace the last paragraph on page 19 with the following rewritten paragraph:

FIGURE 16 illustrates an architecture for the detail tracking layer 541. Each tracking feature provides tracking throughout the duration of a call. A database 1600 such as a relational database, queues tracking/logging requests using a queue mechanism 1602, 1604. The tracking/logging requests are delivered by the queue receiver 1604 and are provided, via a remote interface 1606, to a tracking object 1608 for storage. In at least one embodiment, the tracking object 1608 is implemented as an Enterprise Java Bean (EJB). A data access object 1610 is used to plug-in various different storage mechanisms to the tracking object 1608. The data access object 1610 is generated by a data access object factory 1612.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the second paragraph on page 21 with the following rewritten paragraph:

FIGURE 4 illustrates that the design methodology platform 2 includes a voice gateway 4. The voice gateway 4, in at least one embodiment, incorporates at least some of the functionality of a distributed voice user interface described below. The voice gateway 4 allows the user of a local device 14 (FIGURE 17) to interact with the device 14 by talking to the device 14.

Please replace the second paragraph on page 22 with the following rewritten paragraph:

The Personalized Dialogs service layer 5 is a group of one or more software components that allow a voice applications developer to incorporate natural language concepts into his product in order to present a more human-like and conversational specific voice user interface. The software components of the Personalized Dialogs service layer 5 implement rules for presenting voice information to a user in order to emulate human dialog. Each of the software components may include various constituents necessary for dialog emulation, such as VoiceXML scripts, .WAV files and audio files that make up the dialog presented to the user, recognition grammars that are loaded into speech recognition components, and software code for manipulating the constituents as needed. For example, the Personalized Dialogs service layer 5 includes an error-trapping component 17. The error trapping component 17 is a random prompt pool component. A specific example of this is the error-handling functionality, which has software logic that provides that prompts are not repeated when an error occurs with user voice input. The error trapping component 17 includes code that might provide, upon an error condition, a prompt to the user that says, "I didn't quite get that." If the error condition is not corrected, instead of repeating the prompt,

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

the error trapping component might then provide a prompt to the user that says, "Could you please repeat your selection?" If the error condition is still not corrected, the error trapping component 17 might then provide a prompt that says, "Well, I'm really not understanding you." By providing a series of distinct error-handling prompts rather than repeating the same prompt, a more conversational dialog is carried on with the user than is provided by other voice interface systems.

Please replace the second paragraph on page 23 with the following rewritten paragraph:

Using the components of the Personalized Dialogs service layer 5, an application designer can design a voice user interface 27 that presents data to the user from an existing data system 6, presenting the information in a verbal format that is personalized to the particular user. For instance, the voice user interface 27 can be designed to obtain attribute information about the user. This information could come directly from the user, in response to prompts, or from another source such as a cookie stored on the user's local device 14 (Figure 2). The voice user interface 27 can also be designed to track historical information among multiple sessions with a user, and even to track historical information during a single user session. Using this attribute and historical data, the components of the Personalized Dialogs service layer 5 provide for personalized interaction with the user. For an example that uses attribute data, the voice user interface programmed by the application designer (using the voice integration platform) speaks the user's name when interacting with the user. Similarly, if the user attribute data shows that the user lives in a certain U.S. city, the voice user interface can deliver local weather information to the user. For an example using historical data across more than one session, consider a voice user interface between a user and a data system 6 that provides banking services and data. If the voice user interface 27

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

tracks historical information that indicates that a user, for 10 out of 11 previous sessions (whether conducting the session using a voice interface or another interface such as a GUI), requested a checking account balance upon initiating the session, then the Personalized Dialogs service layer 5 provides for offering the checking account balance to the user at the beginning of the session, without requiring that the user first request the data.

Please replace the third paragraph on page 23 with the following rewritten paragraph:

The Personalized Dialogs service layer 5 also provides for tracking other historical data and using that data to personalize dialogs with the user. For instance, the service layer 5 can be utilized by the application programmer to provide for tracking user preference data regarding advertisements presented to the user during a session. For instance, in at least one embodiment the voice integration platform 2 provides for presenting voice advertisements to the user. The Personalized Dialogs service layer 2 keeps track of user action regarding the advertisements. For instance, a voice ad might say, "Good Morning, Joe, welcome to Global Bank's online service voice system. Would you like to hear about our new money market checking account?" The Personalized Dialogs service layer 5 provides a component that ensures that the format of the ad is rotated so that the wording is different during different sessions. For instance, during a different session the ad might say, "Have you heard about our new money market checking account?" The Personalized Dialog service layer contains a component that provides for tracking how many times a user has heard the advertisement and tracks the user's historical responses to the advertisement. To track the effectiveness of the ad, the Personalized Dialogs service layer 5 keeps track of how many users opt to hear more information about the advertised feature. By tracking user responses to various ads, user preference information is obtained. This historical user preference information is forwarded to the data system 6. Likewise, the Personalized Dialogs service layer 5 has access to

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

historical and attribute data concerning a user that has been stored on the data system 6. This data may come from any of several points of interaction, or "touchpoints", between the user and the data system 6, including telephone access to a staffed call center, voice or non-voice interaction with the data system 6 from a local device such as a personal computer or wireless device, and voice or non-voice telephone communications. This historical user preference information is also maintained for use by the Personalized Dialogs service layer 5. The historical user preference information, along with preference information from the data system 6 that has been obtained during the user's non-voice interaction with the data system 6, is used to provide personalized dialogs to the user and to target specific preference-responsive information to the user.

Please replace the first paragraph on page 25 with the following rewritten paragraph:

The Infrastructure service layer 7 is a group of one or more software components that are necessary for all specific voice user interfaces 27 developed using the voice integration platform 2. For instance, the Infrastructure service layer 7 includes a domain controller software component 15. The domain controller software component 15, also sometimes referred to as a dialog manager, manages and controls the organization and storage of information into logically distinct storage categories referred to herein as "domains". For instance, "electronic mail," "sports scores," "news," and "stock quotes" are examples of four different domains. The domain controller software component 15 provides for storage and retrieval of voice data in the appropriate domain. In some instances, a piece of voice data may be relevant to more than one domain. Accordingly, the domain controller software component 15 provides for storage of the voice data in each of the appropriate domains. The domain controller also traverses the stored domain data to retrieve user-specified data of interest.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the first paragraph on page 26 with the following rewritten paragraph:

The Content Management service layer 11 also contains one or more software components that provide for enhanced management of audio content. For instance, some audio files are streamed from a service to the data system in broad categories. An example of this is the streaming of news and sports headlines to the data system 6 from the Independent Television News ("ITN") network. A content management software component parses the stream of audio content to define constituent portions of the stream. The content management software module then associates each defined constituent portion with a particular domain. For instance, a sports feed can be parsed into college sports and professional sports items that are then associated with the appropriate domain. For smaller granularity, the college sports items are further parsed and associated with football, baseball, basketball, and soccer domains. In this manner the content management software component provides smaller granularity on content than is provided as a streamed audio feed. One skilled in the art will understand that various types of audio data can be received by a data system 6, including voicemail, weather information, stock quotes, and email messages that have been converted to speech. Therefore, the example concerning sports and news headlines audio feed should not be taken to be limiting.

Please replace the third paragraph on page 26 with the following rewritten paragraph:

In at least one embodiment, a content management software component provides templates for the creation of dialogs in a specific voice user interface 27. This feature speeds the creation of dialogs and provides a pre-tested environment for dialog creation that ensures that related components, such as recognition grammars and audio files, are integrated properly.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Please replace the fourth paragraph on page 31 with the following rewritten paragraph:

Remote system 12 may be in communication with the "Internet," thus providing access thereto for users at local devices 14. The Internet is an interconnection of computer "clients" and "servers" located throughout the world and exchanging information according to Transmission Control Protocol/Internet Protocol (TCP/IP), Internetwork Packet eXchange/Sequence Packet eXchange (IPX/SPX), AppleTalk, or other suitable protocol. The Internet supports the distributed application known as the "World Wide Web." Web servers may exchange information with one another using a protocol known as hypertext transport protocol (HTTP). Information may be communicated from one server to any other computer using HTTP and is maintained in the form of web pages, each of which can be identified by a respective uniform resource locator (URL). Remote system 12 may function as a client to interconnect with Web servers. The interconnection may use any of a variety of communication links, such as, for example, a local telephone communication line or a dedicated communication line. Remote system 12 may comprise and locally execute a "web browser" or "web proxy" program. A web browser is a computer program that allows remote system 12, acting as a client, to exchange information with the World Wide Web. Any of a variety of web browsers are available, such as NETSCAPE NAVIGATOR from AOL Time Warner Inc. of New York, NY, INTERNET EXPLORER from Microsoft Corporation of Redmond, WA, and others that allow users to conveniently access and navigate the Internet. A web proxy is a computer program which (via the Internet) can, for example, electronically integrate the systems of a company and its vendors and/or customers, support business transacted electronically over the network (i.e., "e-commerce"), and provide automated access to Web-enabled resources. Any number of web proxies are available, such as B2B INTEGRATION SERVER from webMethods of Fairfax, VA, and MICROSOFT PROXY SERVER from Microsoft Corporation of Redmond, WA. The hardware, software, and

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

protocols--as well as the underlying concepts and techniques--supporting the Internet are generally understood by those in the art.

In the Claims

Please add Claims 25 – 35 as follows:

25. (NEW) A method of designing a voice application, comprising:
designing a dialog ;
coding voice application software to invoke a dialog component;
defining one or more personality rules;
electing to proceed with one of a plurality of grammar development phases, the plurality of grammar development phases including a standard grammar development phase and a custom grammar development phase;
developing a grammar in accordance to the elected grammar development phase;
electing to proceed with one of a plurality of prompt development phases, the plurality of prompt development phases including a standard prompts phase and a speech prompts recording phase;
developing at least one prompt in accordance with the elected prompt development phase;
performing integration to create a specific voice application;
testing the specific voice application; and
deploying the specific voice application.

26. (NEW) The method as recited in Claim 25, wherein:
designing a dialog further comprises definition of a prompt that is designed to elicit a predictable response from a user.

27. (NEW) The method as recited in Claim 25, wherein:
designing a dialog further comprises generating a dialog flow.

28. (NEW) The method as recited in Claim 27, wherein:
designing a dialog further comprises developing the dialog flow into a script.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

29. (NEW) The method as recited in Claim 25, wherein:

coding voice application software to invoke a dialog component further comprises developing software code that invokes the dialog component from a deployment environment.

30. (NEW) The method as recited in Claim 25, wherein:

defining one or more personality rules further comprises defining one or more personality rules in order to impart desired personality features to the specific voice application.

31. (NEW) The method as recited in Claim 25, wherein:

developing a grammar further comprises developing a customized grammar.

32. (NEW) The method as recited in Claim 25, wherein:

developing a grammar further comprises developing the grammar using standard pre-generated grammar components.

33. (NEW) The method as recited in Claim 25, wherein:

developing at least one prompt further comprises generating software code for invoking a predesigned generic prompt.

34. (NEW) The method as recited in Claim 25, wherein:

the custom grammar development phase further comprises recording a custom prompt.

35. (NEW) The method as recited in Claim 34, wherein:

developing at least one prompt further comprises generating software code that is operable to invoke a custom prompt.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

REMARKS

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned "VERSION WITH MARKINGS TO SHOW CHANGES MADE."

The application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned at (512) 794-3600.

EXPRESS MAIL LABEL NO:

EI 830058981 US

Respectfully submitted,

Shireen Irani Bacon
Attorney for Applicant(s)
Reg. No. 40,494

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Specification

The first paragraph on page 1 has been amended as follows:

The present invention relates generally to voice applications and, more particularly, to a methodology of using a **[voice interface]design methodology** platform, having generic software components, and a deployment environment to develop and deploy a specific voice application.

The fourth paragraph on page 2 has been amended as follows:

After the script is developed, a grammar is developed during a grammar development phase. In voice applications that do not use **[DTMF]Dual Tone Multi-Frequency (DTMF)** as the primary caller communication mechanism, Automatic Speech Recognition (ASR) is used to recognize what a caller has spoken and to translate that into the action that the caller wishes to take. The recognition rules for performing these functions are captured within an ASR grammar, which maps a set of allowed utterances to a set of appropriate actions or interpretations. Grammars are specific, i.e., different ASR engines may process rules in different ways.

The last paragraph on page 2 has been amended as follows:

After a grammar is developed then prompts are developed during a prompt development phase. When interacting with a caller, a voice application should speak to the caller. It is known to speak to a caller using **[TTS]Text-to-Speech (TTS)** technology or using pre-recorded speech. In the latter case, a speech talent records phrases that are required as defined within a dialog design specification. It is usual for these prompts to be recorded in an audio format that is not optimized for the telephony environment. Often processing is used to convert the audio format. This conversion is generally performed by a sound processing

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

engine. Also, audio files are often concatenated to build full phrases used by an application as independently recording every utterance would be inefficient.

The third paragraph on page 3 has been amended as follows:

Once tested then the voice application is ready to be deployed. Applications **[are]** must often be deployed on multiple platforms. For example, parts of an application may be deployed to a script server, a voice gateway, an ASR server, etc. Often when deployed, the application environment is replicated for redundancy or scalability.

The third paragraph on page 5 has been amended as follows:

FIGURE 21 is a flow diagram of an exemplary method 200 of operation for **a** remote system.

The last paragraph on page 6 has been amended as follows:

Referring to **[Figure]FIGURE** 1, a voice application is developed in accordance with a design methodology and, in the preferred embodiment, is developed using a design methodology platform 2 (FIGURE 4). The design methodology includes a design phase 1102 and a deployment phase 1104. That is, once a specific voice application is developed and tested in the design phase 1102, the voice application is deployed at deployment **[step]phase** 1104. Once the specific voice application is deployed at deployment **[step]phase** 1104, then the specific voice application is maintained in an iterative fashion as illustrated in FIGURE 1. The regular ongoing maintenance improves voice recognition accuracy and usability. The maintenance includes analyzing data from the specific voice application and making adjustments to the dialog flow and grammars to improve the user's overall user experience with the specific voice application.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

The first paragraph on page 7 has been amended as follows:

Referring to FIGURE 2, the design phase 1102 of the methodology includes the utilization of a plurality of generic software components in order to develop a specific voice application. Such generic software components comprise a design methodology platform 2 (FIGURE 4). More specifically, the design phase 1102 of the voice application design methodology includes utilization of various components of the platform 2. The design phase 1102 of the methodology includes a dialog design phase 1202, a voice coding phase 1204, a personalization phase 1206, a custom prompt development phase 1208, a custom grammar development phase 1210, a standard prompt phase 1212, a standard grammar phase 1214, **and** a system test phase 1216.

The second paragraph on page 7 has been amended as follows:

The dialog design phase 1202 involves designing how the user is expected to interact with the voice application that will be deployed. For example, the dialog design phase 1202 may include the definition of prompts that are designed to elicit a predictable response from the user. Dialog flows may be generated, the dialog flows capturing and defining the anticipated scenarios of user interaction with the voice application that is to be deployed. These dialog flows are developed into scripts. In this manner, during the dialog design phase 1202, a voice application **[user]developer** defines the dialog for the specific voice application under development.

The second paragraph on page 8 has been amended as follows:

In the prompt development phases 1208, 1212, the developer of the voice application **[to]** generates utterances, or prompts, that will be spoken by the voice application to the user. These prompts can be generated using Text-to-speech (TTS) technology and can also be generated using pre-recorded speech. For a voice application that is generated using

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

predesigned generic prompts, phase 1212 is executed. In contrast, custom prompts can be generated in the speech prompts recording phase 1208. In phase 1212, generic pre-generated prompts are used. In phase 1208, a speech talent records prompts as defined by the scripts generated in the dialog design phase 1202.

The third paragraph on page 8 has been amended as follows:

The design methodology also includes grammar development phases 1210, 1214, wherein the developer of the voice application generates grammars. Grammars define which user utterances the voice application recognizes, and provides that the user utterances be translated into a determination of what action the user would like to invoke in the voice application. A grammar is a set of recognition rules. A grammar maps a set of allowed utterances to a set of appropriate actions (or interpretations). One or more grammars for a voice application may be generated using the standard grammar component**s** **in phase** 1214. Similarly, one or more grammars for a voice application may be generated using **[the]** custom grammar development **[component]****in phase** 1210.

The third paragraph on page 9 has been amended as follows:

FIGURES 2 and 3 illustrate**[s]** that generic components 310 are available for the voice application developer's use during the standard prompts 1212 and standard grammar 1214 phases of the design phase 1102. In contrast, custom prompts and grammars are created in phases 1208 and 1210 following the method illustrated in operations 312 through 316 in FIGURE 3. The custom components are coded by the voice application designer in operation 312. The custom-coded components are then tested in a standalone environment in operation 314. If tested successfully, the custom-coded components are then integrated with the voice application. In operation 318, the completed application is tested, regardless of whether the application used standard components or custom-coded components.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

The sixth paragraph on page 14 has been amended as follows:

The component data access layer 810 is an optional part of the component 540, dependent on the functionality required, and is the interface to the back-end resources that are required by the component 540. The component access data layer 810 utilizes the voice application services **layer** 3 to retrieve information from the various back-end data sources and services.

The last paragraph on page 14 has been amended as follows:

FIGURE 9 illustrates a component load sequence. **[Figure]FIGURE** 9 illustrates that a component is invoked in sequence 902 when a request is issued by the voice gateway 4. The request is passed to the dialog control 525 with the URI of the component 540 to be loaded. The dialog control 525 performs a look-up in the dialog control configuration file 530. The dialog control 525 then fetches the appropriate component 540 from the component definition file 802 in sequence 906 and loads the appropriate component in sequence 908. FIGURE 9 illustrates that, in the remaining sequences 909-942, the dialog control 525 forwards the HTTP request to the component 540, passing the parameters from the request. The component script **[504]804** then performs the function that is required by the application, including accessing any back-end services that are required.

The first paragraph on page 15 has been amended as follows:

FIGURE 10 illustrates the architecture of the prompt engine 1000 and associated prompt component. In addition to dialog components 540 the deployment environment 500 supports the invocation of prompt components 1108 by the voice application designer during phase 1212. These are reusable snippets of dynamic code that produce the prompting that is used to communicate with the caller. Each prompt component 1108 is defined and stored within the voice application repository 600 and can be executed by passing a set of input

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

parameters. Prompts are defined within a file, such as an XML file. FIGURE 10 illustrates that, in at least one embodiment, a runtime prompt engine 1000 loads the prompt file from the repository 600 and executes the prompt. **[FIGURE 10 illustrates the architecture of the prompt engine 1000 and associated prompt component.]**

The first paragraph on page 16 has been amended as follows:

FIGURE 5 illustrates that the messaging services layer 539 allows voice applications 27 to send message requests to external systems 542a, 542b, 542n. In at least one embodiment, the messaging services layer 539 is a set of classes that provide the capability to create and bind data messages and send them to specific destinations through utilization of a number of various protocols. The messaging services provided by the messaging services layer **539** are accessible via the voice application service layer 3. The voice application service layer 3 provides a set of custom tag libraries that provide for interaction with the messaging services from within the voice application 27 (FIGURE 4). In at least one embodiment, the voice application service layer 3 provides a tag library to support the following message protocols: Java™ messaging service ("JMS") and Java™ XML messaging service ("JAXM").

The last paragraph on page 17 has been amended as follows:

In such cases (i.e., when a message is expired) , the messaging controller 1220 performs data-expiration monitoring and cleanup. If the application 27 requests to receive a message and there is no message **[avaialbe]available**, then the messaging **[controler]controller** 1220 notifies the application 27 to wait for a specified timeout period. If the requested message does not become available during the timeout period, then the request becomes "timed-out." In such case, the messaging controller 1220 provides a negative acknowledgement to the message source 542 to indicate the requested message has not been delivered.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

The second paragraph on page 18 has been amended as follows:

FIGURE 5 illustrates that another set of custom tags is provided to provide access to a rules integration layer 537. This set of tags, referred to as a personalization tag library, allows rules to be set up in a rules engine associated with the rules integration layer 537. This allows a rules set to be invoked and also allows for actions to be taken in the dialog based on the result of invocation of the rules set. The tags in the personalization tag library also provide access to information being **[racked]**<u>**tracked**</u> by the application. The custom tags allow the application developer to execute a rule set on a number of parameters to decide what actions should be performed by the application. Some appropriate uses for this interface are:

The first paragraph on page 19 has been amended as follows:

FIGURE 15 illustrates the architecture for the rules integration layer 537. The rules integration layer 537 includes a remote interface 1502 that facilitates communication with the voice application services layer 3. The rules integration layer <u>**537**</u> includes a set of custom tags that perform specific rule-type functions and includes a rules engine 1504. A rule engine pool manager 1506 **[that]** performs initialization functions by polling and sharing available rule engine resources. Remote users assert objects into the context of the rule engine 1504 and then execute one or more rule files 1508. The rules service remote interface 1502 maintains state across method calls, allowing multiple objects to be asserted before rules are invoked. Once the rules have been executed, the rules integration layer 537 returns all objects that were affected by the execution of the rules. In at least one embodiment, the affected objects are returned as an object array.

The third paragraph on page 19 has been amended as follows:

FIGURE 5 illustrates that another set of custom tags is provided to provide access to a call detail tracking layer 541. The custom tags, referred to as a tracking tag library, provide for tracking of information **[bout]**<u>**about**</u> a caller session while the application is executing.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

To do this, the call detail tracking layer 541 provides for tracking information falling into a number of distinct categories, including call-based information, caller-based information, and event-based information. Call-based tracking involves tracking information about a particular call, including when the call started, dialed number, etc. Caller-based tracking involves tracking information about caller. This tracking feature is used in conjunction with a user profile in order to provide personalized content to the caller. Event-based tracking involves tracking the events that occur within the call flow. For example, event-based tracking will record that a caller has been transferred out of the specific voice application. As another example, event-based tracking will record that a caller has provided an invalid credential.

The last paragraph on page 19 has been amended as follows:

[Figure]<u>FIGURE</u> 16 illustrates an architecture for the detail tracking layer 541. Each tracking feature provides tracking throughout the duration of a call. A database 1600 such as a relational database, queues tracking/logging requests using a queue mechanism 1602, 1604. The tracking/logging requests are delivered by the queue receiver 1604 and are provided, via a remote interface 1606, to a tracking object 1608 for storage. In at least one embodiment, the tracking object 1608 is implemented as an Enterprise Java Bean (EJB). A data access object 1610 is used to plug-in various different storage mechanisms to the tracking object 1608. The data access object 1610 is generated by a data access object factory 1612.

The second paragraph on page 21 has been amended as follows:

FIGURE 4 illustrates that the design methodology platform 2 includes a voice gateway 4. The voice gateway 4, in at least one embodiment, incorporates at least some of the functionality of a distributed voice user interface described below. The voice gateway 4 allows the user of a local device 14 **[(Figure 2)]**<u>(FIGURE 17)</u> to interact with the device 14 by talking to the device 14.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

The second paragraph on page 22 has been amended as follows:

The Personalized Dialogs service layer 5 is a group of one or more software components that allow a voice applications developer to incorporate natural language concepts into his product in order to present a more human-like and conversational specific voice user **[interface .]**__interface.__ The software components of the Personalized Dialogs service layer 5 implement rules for presenting voice information to a user in order to emulate human dialog. Each of the software components may include various constituents necessary for dialog emulation, such as **[voice XML]**__VoiceXML__ scripts, .WAV files and audio files that make up the dialog presented to the user, recognition grammars that are loaded into speech recognition components, and software code for manipulating the constituents as needed. For example, the Personalized Dialogs service layer 5 includes an error-trapping component 17. The error trapping component 17 is __a random prompt pool component. A specific example of this is the error-handling functionality, which has__ software logic that provides that prompts are not repeated when an error occurs with user voice input. The error trapping component 17 includes code that might provide, upon an error condition, a prompt to the user that says, "I didn't quite get that." If the error condition is not corrected, instead of repeating the prompt, the error trapping component might then provide a prompt to the user that says, "Could you please repeat your selection?" If the error condition is still not corrected, the error trapping component 17 might then provide a prompt that says, "Well, I'm really not understanding you." By providing a series of distinct error-handling prompts rather than repeating the same prompt, a more conversational dialog is carried on with the user than is provided by other voice interface systems.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

The second paragraph on page 23 has been amended as follows:

Using the components of the Personalized Dialogs service layer 5, an application designer can design a voice user interface 27 that presents data to the user from an existing data system 6, presenting the information in a verbal format that is personalized to the particular user. For instance, the voice user interface 27 can be designed to obtain attribute information about the user. This information could come directly from **the** user, in response to prompts, or from another source such as a cookie stored on the user's local device 14 (Figure 2). The voice user interface 27 can also be designed to track historical information among multiple sessions with a user, and even to track historical information during a single user session. Using this attribute and historical data, the components of the Personalized Dialogs service layer 5 provide for personalized interaction with the user. For an example that uses attribute data, the voice user interface programmed by the application designer (using the voice integration platform) speaks the user's name when interacting with the user. Similarly, if the user attribute data shows that the user lives in a certain U.S. city, the voice user interface can deliver local weather information to the user. For an example using historical data across more than one session, consider a voice user interface between a user and a data system 6 that provides banking services and data. If the voice user interface 27 tracks historical information that indicates that a user, for 10 out of 11 previous sessions (whether conducting the session using a voice interface or another interface such as a GUI), requested a checking account balance upon initiating the session, then the Personalized Dialogs service layer 5 provides for offering the checking account balance to the user at the beginning of the session, **[without requiring]without requiring** that the user first request the data.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

The third paragraph on page 23 has been amended as follows:

The Personalized Dialogs service layer 5 also provides for tracking other historical data and using that data to personalize dialogs with the user. For instance, the service layer 5 can be utilized by the application programmer to provide for tracking user preference data regarding advertisements presented to the user during a session. For instance, in at least one embodiment the voice integration platform 2 provides for presenting voice advertisements to the user. The Personalized Dialogs service layer 2 keeps track of user action regarding the advertisements. For instance, a voice **[add]**<u>**ad**</u> might say, "Good Morning, Joe, welcome to Global Bank's online service voice system. Would you like to hear about our new money market checking account?" The Personalized Dialogs service layer 5 provides a component that ensures that the format of the ad is rotated so that the wording is different during different sessions. For instance, during a different session the ad might say, "Have you heard about our new money market checking account?" The Personalized Dialog service layer contains a component that provides for tracking how many times a user has heard the advertisement and tracks the user's historical responses to the advertisement. To track the effectiveness of the **[add]**<u>**ad**</u>, the Personalized Dialogs service layer 5 keeps track of how many users opt to hear more information about the advertised feature. By tracking user responses to various **[adds]**<u>**ads**</u>, user preference information is obtained. This historical user preference information is forwarded to the data system 6. Likewise, the Personalized Dialogs service layer 5 has access to historical and attribute data concerning a user that has been stored on the data system 6. This data may come from any of several points of interaction, or "touchpoints", between the user and the data system 6, including telephone access to a **[manned]**<u>**staffed**</u> call center, voice or non-voice interaction with the data system 6 from a local device such as a personal computer or wireless device, and voice or non-voice telephone communications. This historical user preference information is also maintained for use by the

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

Personalized Dialogs service layer 5. The historical user preference information, along with preference information from the data system 6 that has been obtained during the user's non-voice interaction with the data system 6, is used to provide personalized dialogs to the user and to target specific preference-responsive information to the user.

The first paragraph on page 25 has been amended as follows:

The Infrastructure service layer 7 is a group of one or more software components that are necessary for all specific voice user interfaces 27 developed using the voice integration platform 2. For instance, the Infrastructure service layer 7 includes a domain controller software **[component 15]**__component 15__. The domain controller software component 15, __also sometimes referred to as a dialog manager,__ manages and controls the organization and storage of information into logically distinct storage categories referred to herein as "domains". For instance, **["electronic mail", "sports scores" and "news"]**__"electronic mail," "sports scores," "news," and__ "stock quotes" are examples of four different domains. The domain controller software component 15 provides for storage and retrieval of voice data in the appropriate domain. In some instances, a piece of voice data may be relevant to more than one domain. Accordingly, the domain controller software component 15 provides for storage of the voice data in each of the appropriate domains. The domain controller also traverses the stored domain data to retrieve user-specified data of interest.

The first paragraph on page 26 has been amended as follows:

The Content Management service layer 11 also contains one or more software components that provide for enhanced management of audio content. For instance, some audio files are streamed from a service to the data system in broad categories. An example of this is the streaming of news and sports headlines to the data system 6 from the Independent Television News __("ITN")__ network. A content management software component parses the

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

stream of audio content to define constituent portions of the stream. The content management software module then associates each defined constituent portion with a particular domain. For instance, a sports feed can be parsed into college sports and professional sports items that are then associated with the appropriate domain. For smaller granularity, the college sports items are further parsed and associated with football, baseball, basketball, and soccer domains. In this manner the content management software component provides smaller granularity on content than is provided as a streamed audio feed. One skilled in the art will understand that various types of audio data can be received by a data system 6, including voicemail, weather information, stock quotes, and email messages that have been converted to speech. Therefore, the example concerning sports and news headlines audio feed should not be taken to be limiting.

The third paragraph on page 26 has been amended as follows:

In at least one embodiment, a content management software component provides templates for the creation of dialogs in a specific voice user interface 27. This feature speeds the creation of dialogs and provides a pre-tested environment for dialog creation that ensures that related components, such as recognition grammars and **[.wav]**<u>**audio**</u> files, are integrated properly.

The fourth paragraph on page 31 has been amended as follows:

Remote system 12 may be in communication with the "Internet," thus providing access thereto for users at local devices 14. The Internet is an interconnection of computer "clients" and "servers" located throughout the world and exchanging information according to Transmission Control Protocol/Internet Protocol (TCP/IP), Internetwork Packet eXchange/Sequence Packet eXchange (IPX/SPX), AppleTalk, or other suitable protocol. The Internet supports the distributed application known as the "World Wide Web." Web servers

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

may exchange information with one another using a protocol known as hypertext transport protocol (HTTP). Information may be communicated from one server to any other computer using HTTP and is maintained in the form of web pages, each of which can be identified by a respective uniform resource locator (URL). Remote system 12 may function as a client to interconnect with Web servers. The interconnection may use any of a variety of communication links, such as, for example, a local telephone communication line or a dedicated communication line. Remote system 12 may comprise and locally execute a "web browser" or "web proxy" program. A web browser is a computer program that allows remote system 12, acting as a client, to exchange information with the World Wide Web. Any of a variety of web browsers are available, such as NETSCAPE NAVIGATOR from **[Netscape Communications Corp.]**<u>**AOL Time Warner Inc.**</u> of **[Mountain View, CA]**<u>**New York, NY**</u>, INTERNET EXPLORER from Microsoft Corporation of Redmond, WA, and others that allow users to conveniently access and navigate the Internet. A web proxy is a computer program which (via the Internet) can, for example, electronically integrate the systems of a company and its vendors and/or customers, support business transacted electronically over the network (i.e., "e-commerce"), and provide automated access to Web-enabled resources. Any number of web proxies are available, such as B2B INTEGRATION SERVER from webMethods of Fairfax, VA, and MICROSOFT PROXY SERVER from Microsoft Corporation of Redmond, WA. The hardware, software, and protocols--as well as the underlying concepts and techniques--supporting the Internet are generally understood by those in the art.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

<u>In the Claims</u>

Claims 25 - 35 have been added as follows:

25. (NEW) A method of designing a voice application, comprising:

designing a dialog ;

coding voice application software to invoke a dialog component;

defining one or more personality rules;

electing to proceed with one of a plurality of grammar development phases, the plurality of grammar development phases including a standard grammar development phase and a custom grammar development phase;

developing a grammar in accordance to the elected grammar development phase;

electing to proceed with one of a plurality of prompt development phases, the plurality of prompt development phases including a standard prompts phase and a speech prompts recording phase;

developing at least one prompt in accordance with the elected prompt development phase;

performing integration to create a specific voice application;

testing the specific voice application; and

deploying the specific voice application.

26. (NEW) The method as recited in Claim 25, wherein:

designing a dialog further comprises definition of a prompt that is designed to elicit a predictable response from a user.

27. (NEW) The method as recited in Claim 25, wherein:

designing a dialog further comprises generating a dialog flow.

28. (NEW) The method as recited in Claim 27, wherein:

designing a dialog further comprises developing the dialog flow into a script.

29. (NEW) The method as recited in Claim 25, wherein:

coding voice application software to invoke a dialog component further comprises developing software code that invokes the dialog component from a deployment environment.

30. (NEW) The method as recited in Claim 25, wherein:

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

defining one or more personality rules further comprises defining one or more personality rules in order to impart desired personality features to the specific voice application.

31. (NEW) The method as recited in Claim 25, wherein:
developing a grammar further comprises developing a customized grammar.

32. (NEW) The method as recited in Claim 25, wherein:
developing a grammar further comprises developing the grammar using standard pre-generated grammar components.

33. (NEW) The method as recited in Claim 25, wherein:
developing at least one prompt further comprises generating software code for invoking a predesigned generic prompt.

34. (NEW) The method as recited in Claim 25, wherein:
the custom grammar development phase further comprises recording a custom prompt.

35. (NEW) The method as recited in Claim 34, wherein:
developing at least one prompt further comprises generating software code that is operable to invoke a custom prompt.

LAW OFFICES OF
SKJERVEN MORRILL
MACPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979